Amdocs training – 24 July to 8 August 2025

Day 1 – 24 July 2025 – Jenkins & Maven project & Pipeline deployment.

Day 2 – 25 July 2025 – Cloud & SQL, DBMS

- Relational DBs – SQL, My SQL.
- Create, drop, insert, show, select, alter, truncate.
- Procedures in PL/SQL.
- Fields, views, foreign key, primary Key -> their properties.
- Data integrity, normalization and Redundancy.
- Row_Number, Limit, join, union
- Indexes – special lookup table: single column, unique indexes, drop index command

Day 3 – 28 July 2025 - PL/SQL Commands on Oracle ;

- while, for, if-else, data type.
- X … Y (lower bound … upper bound)
- goto, continue (it will start form the beginning)
- end if
- procedures in PL/SQL – useful for storing queries
  - procedures to create table -> insert values into table (id -> number, name –> varchar)
  - CREATE -> IS -> BEGIN -> END
  - Getting error in procedure (try again)

Day 4 – 29 July 2025 - PL/SQL

- Create table students an display table data using procedure
- 4 -Implicit cursors or Explicit cursors in cursor types in PL/SQL.
- Cursor -> is a pointer to a context area.
- Implicit cursors (always closed) - %FOUND, %NOTFOUND, %ISOPEN, %ROWCOUNT.
- **Implicit cursor** example: **Updates customer data and shows how many rows were changed.**

```
declare
    total rows(2);
begin
```

```
Update Customers
Set salary = salary +500;
if sql%notfound then
    dbms_output.put_line('no customers selected');
else if sql%found then
    total_rows:= sql%rowcount;
    dbms_output.put_line(total_rows || 'customers selected');
end if;
end;
```

- This **UPDATE** line **does not actually change any data** — it just reassigns the phone column to itself. It's often used as a placeholder to **trigger an update** without modifying values, just to demonstrate how SQL%ROWCOUNT works.

- Cursor steps:
  - Declare -> deploy-> fetch -> close (the cursor)
  - Cursor example (photo in phone) -> explicit cursor ->
- Triggers -> benefits -> creating triggers
  - Customer table -> create or replace trigger "**display salary changes**"
- Packages in PL/SQL
  - Create package
    - Step 1- Package specification -> simply declaring the package functions/procedures.
    - Step 2- Package body
      - Functions, procedures, triggers etc. are present in package body.
    - Step 3 – How we are going to use the package
      - Package name package element
      - Package_name.procedure_to be use
- Benefits of Package

Day 4 – 29 July 2025 - SECURITY

- Security, threats (external & Internal), DDoS attacks, malware, phishing, data breaches
- Importance of software Security, Issues related to software security
- Tools For Software Security

- Software Security VS Cyber Security
- Best Practices for Software Security -> penetration testing, regular patching, access control, encryption, security training, incident response plan, secure development lifecycle.
- Case Study on Cyber security
- OSI model, Melissa virus, computer worm (ILOVEYOU computer worm), trojan Horse,

Day 5 – 30 July 2025 - GenAI

- AI definition, disadvantages & advantages of AI
- types of AI: Weak AI, strong AI (General AI), Super AI
- Case Study on AI.
- Read about AWS Security like Identity Access Management.
- Encryption Techniques

Day 6 – 31st July 2025 – Python

- Python basics, loops, conditional statements, variables, data types
- Operators in python
- Armstrong numbers code in python
- Even number code in python

Day 7 – 1st August 2025 – Python

- OOPs, objects, classes
- Use of self, __init__ in python
- Use of pass keyword in python
- Use of double and single underscore
- Use of __str__
- Types of constructors -> default, parametrized (imp to read and understand again)
    - Instructor asked it 3-4 times whether we've understood the concept or not.
- OOPS VS POPL
  POPL -> used mainly in C, Pascal while OOPs is generally used in C++ ,Java
- Abstraction in python
- Encapsulation in python
    - Can we achieve encapsulation through abstraction and vice verse ? and if no, then Why ?
- Access modifiers in python: public, private, protected (_p)
    - Object will access the local value present in the class

- Access modifiers in python code

```
class Employee:
  def __init__(self, name, salary, password):
    self.name = name # public attribute
    self._salary=salary # private attribute
    self.__password=password # protected attribute

  def show_details(self):
    print(f"Name: {self.name}")
    print(f"Salary: {self._salary}")
    print(f"Password: {self.__password}") # accessible within class
# creating object
emp=Employee("Jyoti",75000,"secure@123")

# accessing attributes
print("Public: ", emp.name)
print("Protected: ",emp._salary)
# print("Private: ",emp.__password) # will raise attribute error

# accesing private attributes using name mangling
print("Private (via name Mangling): ",emp._Employee__password)
# using method to display all names
emp.show_details();
```

- Inheritance in Python: multiple, single
- Polymorphism types: operator overloading and operator overriding

STUDENT MANAGEMENT SYSTEM -> LIST, TUPES, DICTIONARY, FILE HANDLING, EXCEPTION HANDLING (to Do explore through example)

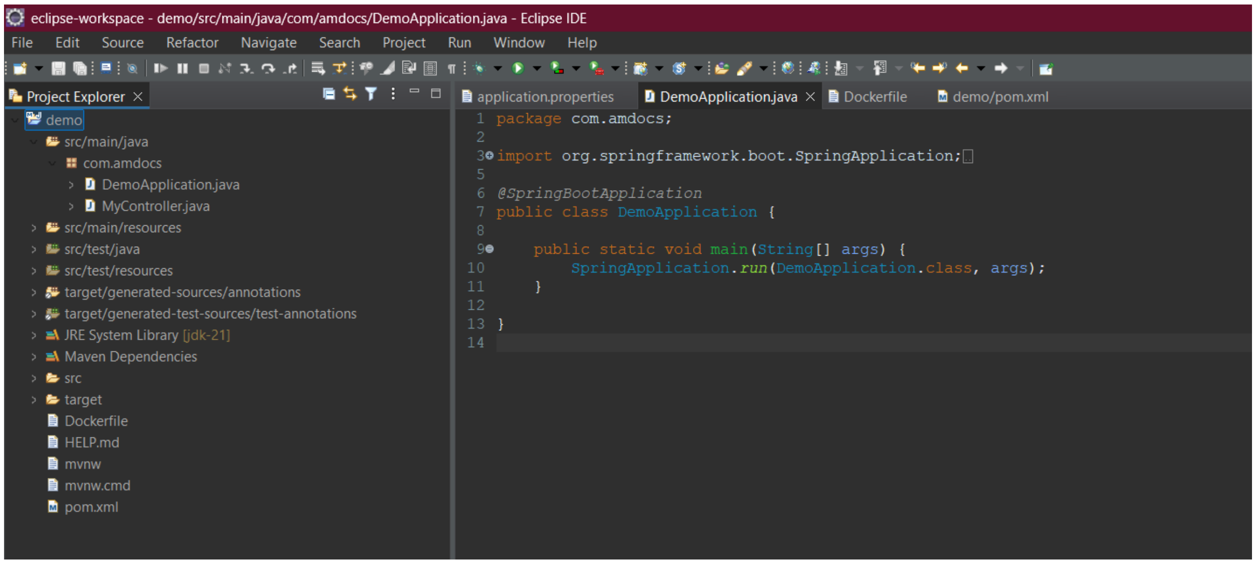
Day 8 – 4th August 2025 – ETL in Python

- Load dataset, see its structure and read csv file using ETL techniques.
- Use of box plot, swarm plot and transform techniques to use data.
- Import dataset and perform ETL
  - Remove duplicates, null and convert column header to lower case.
  - Make a database and Import dataset to mysql workbench using Jupyter/Colab.
  - Can store box plots in AWS S3 bucket also.

- Demo for apache pyspark -> API for Apache spark
- Build heatmap for this.
- Sales Analysis in Pyspark.

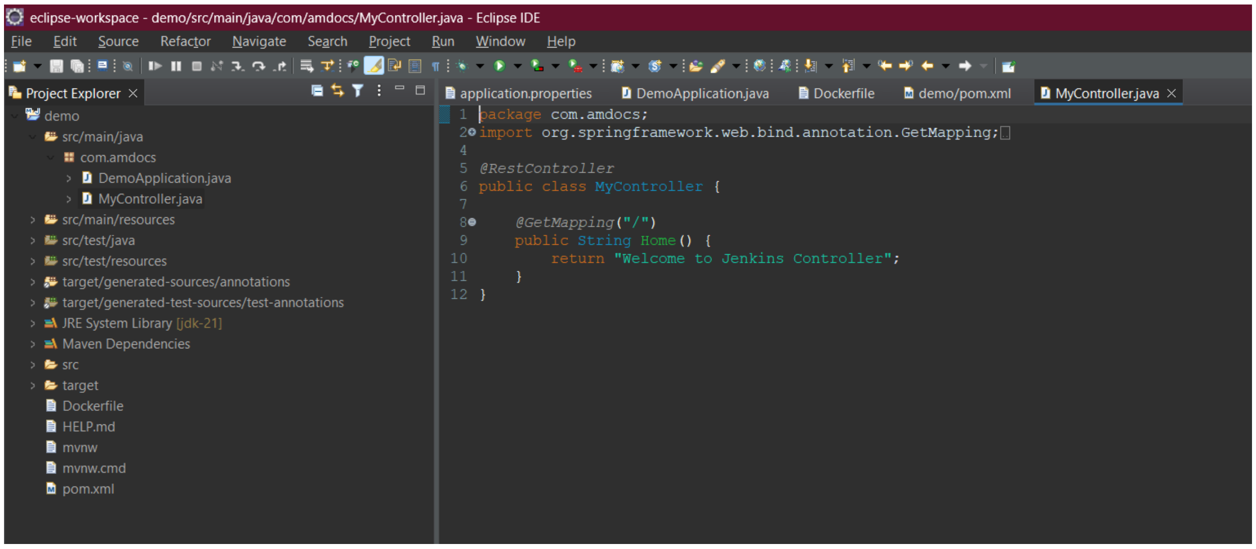
Day 9 – 5th August 2025 – DevSecOps

- What is DevSecOps, why it is needed and how it's beneficial ?
- Jenkins Demo: build project,
- Steps to build a small spring boot project.
    1. Run DemoApplicationJava


eclipse-workspace - demo/src/main/java/com/amdocs/DemoApplication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
demo
src/main/java
com.amdocs
DemoApplication.java
MyController.java
src/main/resources
src/test/java
src/test/resources
target/generated-sources/annotations
target/generated-test-sources/test-annotations
JRE System Library [jdk-21]
Maven Dependencies
src
target
Dockerfile
HELP.md
mvnw
mvnw.cmd
pom.xml
application.properties
DemoApplication.java
Dockerfile
demo/pom.xml

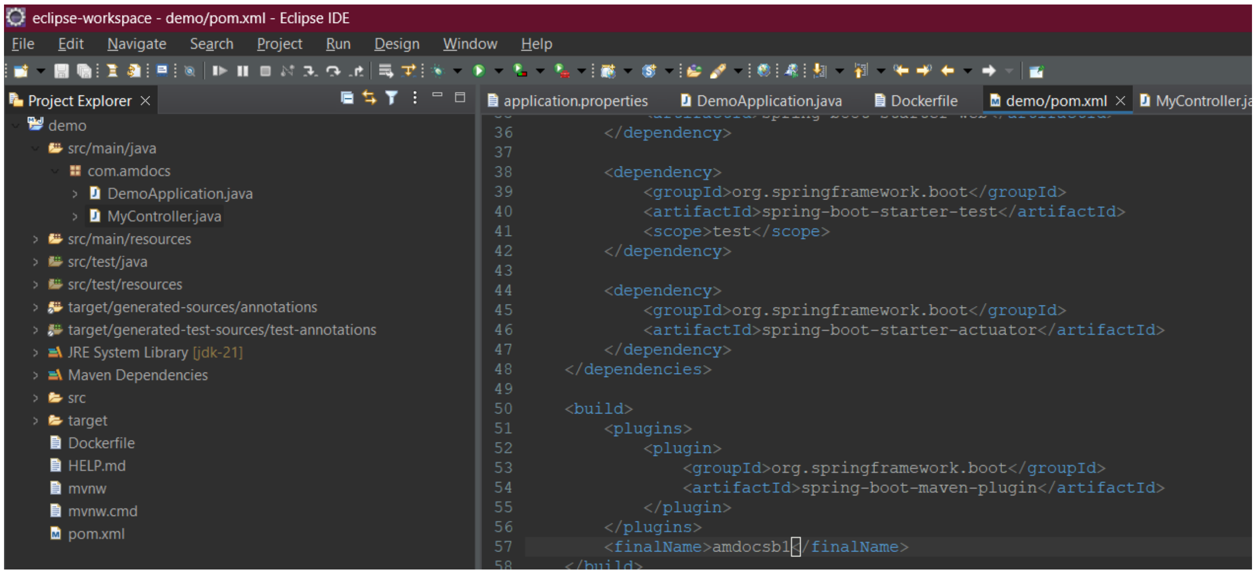
```java
package com.amdocs;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

2. For MyController file



```java
package com.amdocs;
import org.springframework.web.bind.annotation.GetMapping;

@RestController
public class MyController {

    @GetMapping("/")
    public String Home() {
        return "Welcome to Jenkins Controller";
    }
}
```

3. For Springboot starter actuator in demo/pom.xml

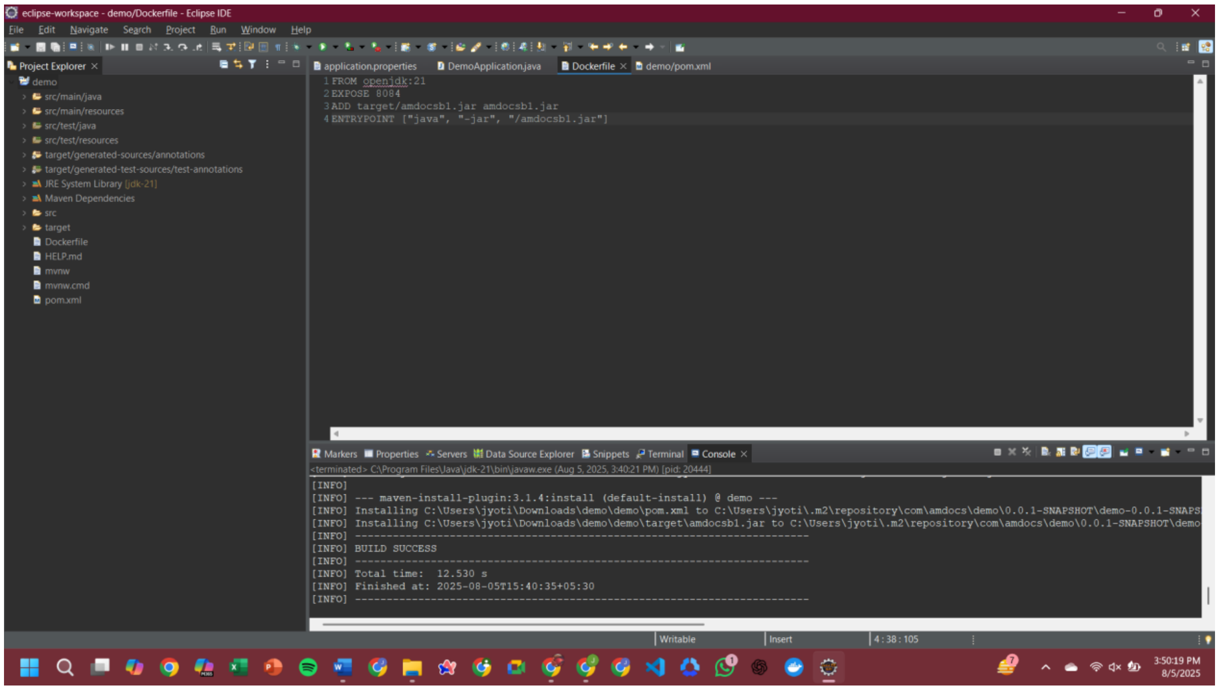


```xml
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
        <finalName>amdocsb1</finalName>
```

- @RestController class will help to create a REST API in pipeline.

STEP TO create, run(build), tag and push images in DOCKER.

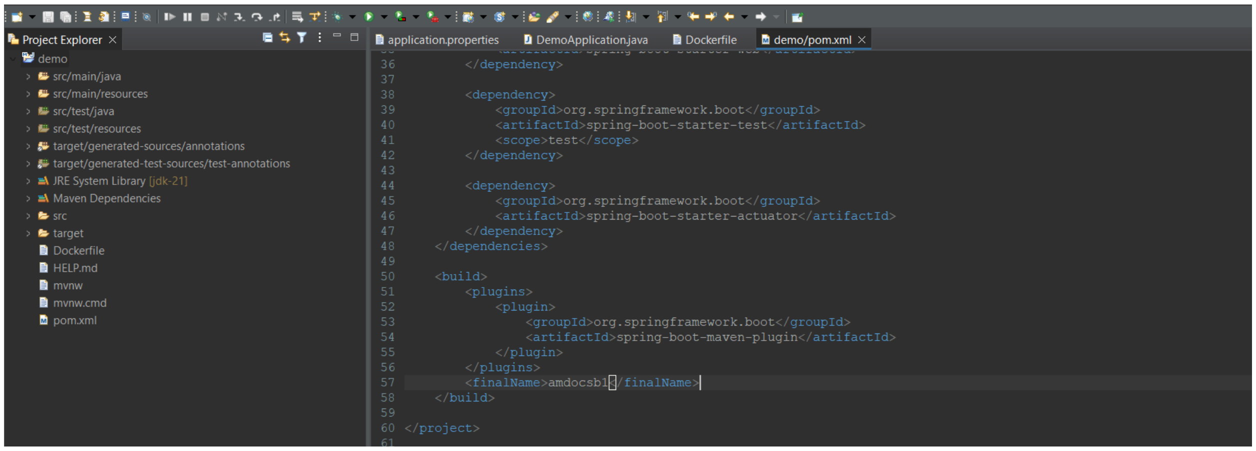
- When you run an image in docker it will become a container.



```
FROM openjdk:21
EXPOSE 8084
ADD target/amdocsb1.jar amdocsb1.jar
ENTRYPOINT ["java", "-jar", "/amdocsb1.jar"]
```

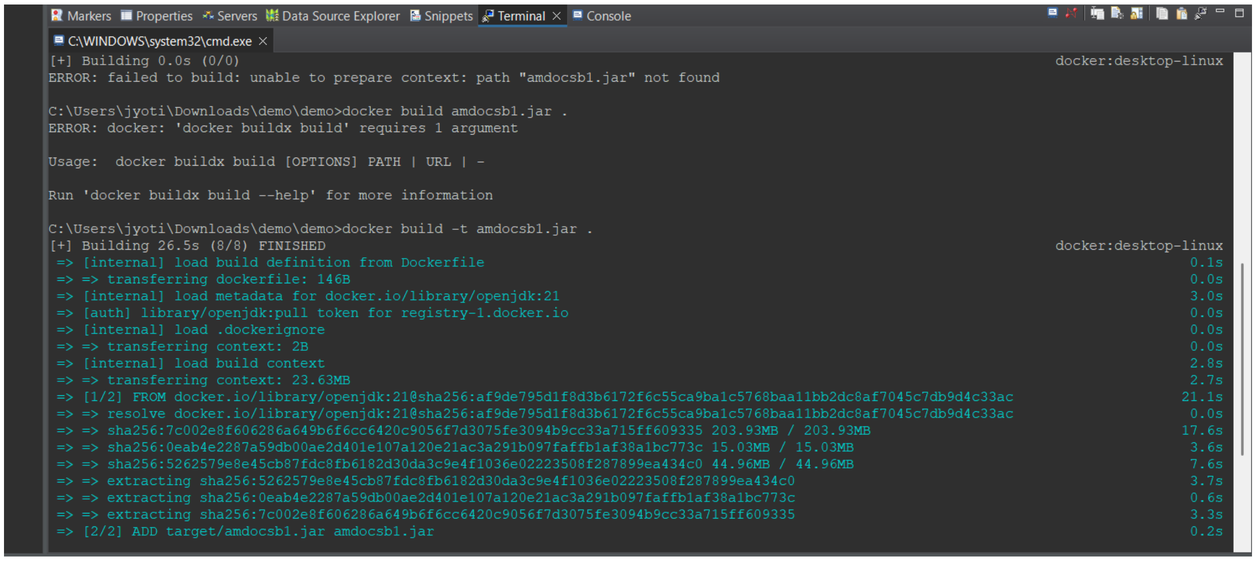
```
<terminated> C:\Program Files\Java\jdk-21\bin\javaw.exe (Aug 5, 2025, 3:40:21 PM) [pid: 20444]
[INFO]
[INFO] --- maven-install-plugin:3.1.4:install (default-install) @ demo ---
[INFO] Installing C:\Users\jyoti\Downloads\demo\demo\pom.xml to C:\Users\jyoti\.m2\repository\com\amdocs\demo\0.0.1-SNAPSHOT\demo-0.0.1-SNAPS
[INFO] Installing C:\Users\jyoti\Downloads\demo\demo\target\amdocsb1.jar to C:\Users\jyoti\.m2\repository\com\amdocs\demo\0.0.1-SNAPSHOT\demo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.530 s
[INFO] Finished at: 2025-08-05T15:40:35+05:30
[INFO] ------------------------------------------------------------------------
```

- Learn how to write a dockerfile.



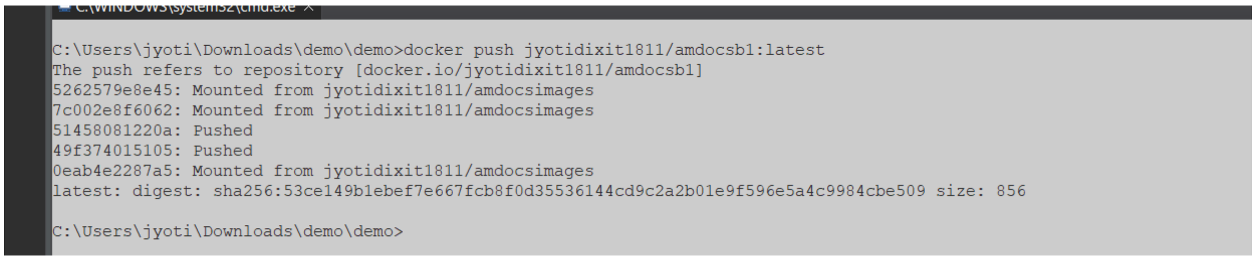
```xml
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
        <finalName>amdocsb1</finalName>
    </build>

</project>
```

- To build own image in docker:
    1. Go back to the project again
    2. Type command: docker build amdocb1.jar . (dot means all)



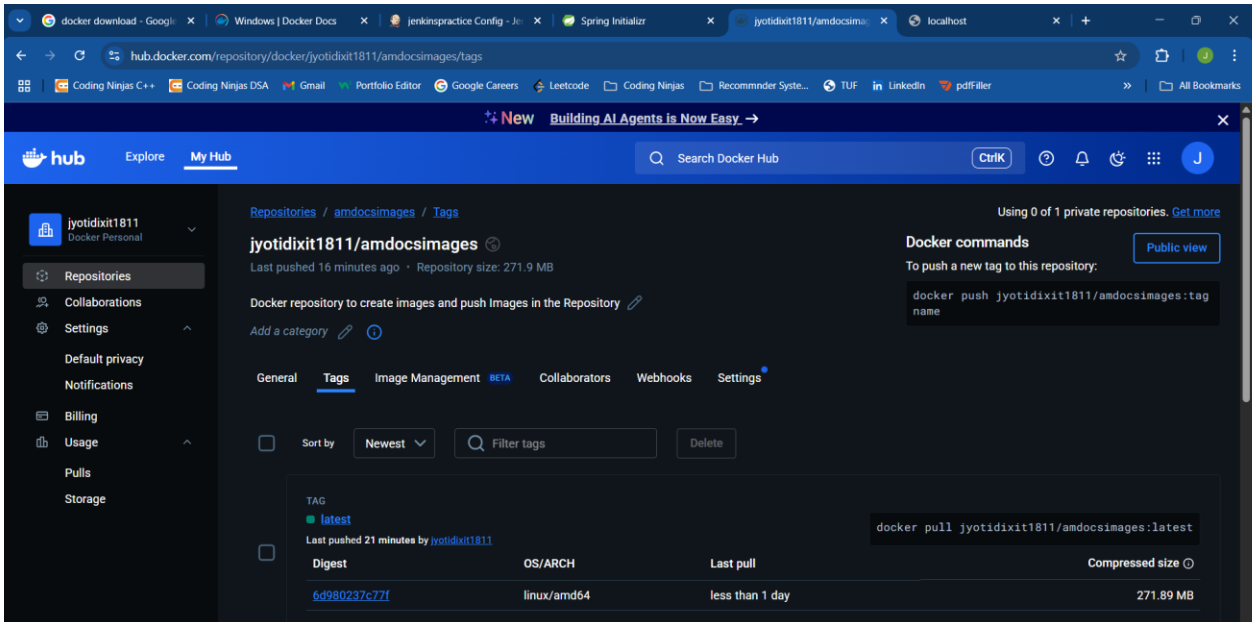
```
[+] Building 0.0s (0/0)                                                                             docker:desktop-linux
ERROR: failed to build: unable to prepare context: path "amdocsb1.jar" not found

C:\Users\jyoti\Downloads\demo\demo>docker build amdocsb1.jar .
ERROR: docker: 'docker buildx build' requires 1 argument

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Run 'docker buildx build --help' for more information

C:\Users\jyoti\Downloads\demo\demo>docker build -t amdocsb1.jar .
[+] Building 26.5s (8/8) FINISHED                                                                   docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 146B                                                                                0.0s
 => [internal] load metadata for docker.io/library/openjdk:21                                                       3.0s
 => [auth] library/openjdk:pull token for registry-1.docker.io                                                      0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build context                                                                                   2.8s
 => => transferring context: 23.63MB                                                                                2.7s
 => [1/2] FROM docker.io/library/openjdk:21@sha256:af9de795d1f8d3b6172f6c55ca9ba1c5768baa11bb2dc8af7045c7db9d4c33ac 21.1s
 => => resolve docker.io/library/openjdk:21@sha256:af9de795d1f8d3b6172f6c55ca9ba1c5768baa11bb2dc8af7045c7db9d4c33ac  0.0s
 => => sha256:7c002e8f606286a649b6f6cc6420c9056f7d3075fe3094b9cc33a715ff609335 203.93MB / 203.93MB                 17.6s
 => => sha256:0eab4e2287a59db00ae2d401e107a120e21ac3a291b097faffb1af38a1bc773c 15.03MB / 15.03MB                    3.6s
 => => sha256:5262579e8e45cb87fdc8fb6182d30da3c9e4f1036e02223508f287899ea434c0 44.96MB / 44.96MB                    7.6s
 => => extracting sha256:5262579e8e45cb87fdc8fb6182d30da3c9e4f1036e02223508f287899ea434c0                           3.7s
 => => extracting sha256:0eab4e2287a59db00ae2d401e107a120e21ac3a291b097faffb1af38a1bc773c                           0.6s
 => => extracting sha256:7c002e8f606286a649b6f6cc6420c9056f7d3075fe3094b9cc33a715ff609335                           3.3s
 => [2/2] ADD target/amdocsb1.jar amdocsb1.jar                                                                      0.2s
```

- Command to push images in docker.



```
C:\Users\jyoti\Downloads\demo\demo>docker push jyotidixit1811/amdocsb1:latest
The push refers to repository [docker.io/jyotidixit1811/amdocsb1]
5262579e8e45: Mounted from jyotidixit1811/amdocsimages
7c002e8f6062: Mounted from jyotidixit1811/amdocsimages
51458081220a: Pushed
49f374015105: Pushed
0eab4e2287a5: Mounted from jyotidixit1811/amdocsimages
latest: digest: sha256:53ce149b1ebef7e667fcb8f0d35536144cd9c2a2b01e9f596e5a4c9984cbe509 size: 856

C:\Users\jyoti\Downloads\demo\demo>
```

- Output of pushing images into Docker.



Repositories / amdocsimages / Tags

jyotidixit1811/amdocsimages

Last pushed 16 minutes ago · Repository size: 271.9 MB

Docker repository to create images and push Images in the Repository

Docker commands

To push a new tag to this repository:

```
docker push jyotidixit1811/amdocsimages:tagname
```

TAG: latest

Last pushed 21 minutes by jyotidixit1811

```
docker pull jyotidixit1811/amdocsimages:latest
```

Digest	OS/ARCH	Last pull	Compressed size
6d980237c77f	linux/amd64	less than 1 day	271.89 MB

Docker main screen -> to show that amocdb1.jar file has been created.

Day 10: Project Evaluation + Viva